

Absolvování individuální odborné praxe

Individual Professional Practice in the Company

Zadání bakalářské práce

Student:

Michal Kania

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

**Absolvování individuální odborné praxe
Individual Professional Practice in the Company**

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: E LINKX a.s.
2. Struktura závěrečné zprávy:
 - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta
 - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti
 - c) Zvolený postup řešení zadaných úkolů
 - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe
 - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe
 - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vedl odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. David Ježek, Ph.D.**

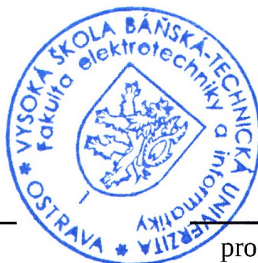
Konzultant bakalářské práce: Ing. Roman Hrdý

Datum zadání: 16.11.2012

Datum odevzdání: 07.05.2013



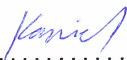
doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 25. dubna 2013

.....

Rád bych na tomto místě poděkoval společnosti E LINKX a.s., především panu Ing. Romanu Hrdému, za cenné rady a odborné vedení v průběhu celé praxe a také panu Ing. Davidu Ježkovi, Ph.D. za konzultace spojené s touto bakalářskou prací.

Abstrakt

V této bakalářské práci jsou popsány projekty na kterých jsem pracoval v rámci bakalářské praxe ve firmě E LINKX a.s.. Ve firmě jsem byl na pozici programátora aplikací. Projekty, které zde budu popisovat vznikly převážně z důvodu uspokojení potřeb zákazníků firmy E LINKX a.s.. Práce bude také obsahovat stručný souhrn znalostí získaných v době studia oboru informatiky a výpočetní techniky na Vysoké škole báňské - Technické univerzitě v Ostravě a jejich uplatnění v praxi.

Klíčová slova: praxe, E LINKX a.s., C#, .NET, aplikace, SQL, HTML, CSS, XML, XSLT

Abstract

In this bachelor thesis, there is a description of projects I was working on during my professional practice in a company named E LINKX a.s.. I was working there as an application programmer. Projects, which are described in this thesis, were created mainly on the basis of satisfying the needs of customers of E LINKX a.s. company. The thesis contains a brief summary of the knowledge acquired during the studies of Computer Science and Technology on VŠB–Technical University of Ostrava and also their application in practice.

Keywords: practice, E LINKX a.s., C#, .NET, application, SQL, HTML, CSS, XML, XSLT

Seznam použitých zkratek a symbolů

ERP	– Enterprise resource planning
B2B	– Business to business
B2C	– Business to customer
DLL	– Dynamic-link library
XSLT	– eXtensible Stylesheet Language Transformations
XML	– eXtensible Markup Language
WUC	– Web User Control
RFC	– Requests for comment
DNS	– Domain Name System
MX	– Mail eXchange record
HTML	– HyperText Markup Language
CSS	– Cascading Style Sheets
RTF	– Rich Text Format
T-SQL	– Transact-Structured Query Language
UML	– Unified Modeling Language
XPath	– XML Path Language

Obsah

1	Úvod	4
2	Popis odborného zaměření firmy a pracovního zařazení studenta	5
2.1	Odborné zaměření firmy	5
2.2	Pracovní zařazení studenta	5
3	Seznam úkolů v průběhu praxe a jejich přibližná časová náročnost	6
3.1	Paragrafy	6
3.2	EmailChecker	6
3.3	XmlViewer	6
3.4	Xslt	7
3.5	Xslt z více souborů	7
3.6	Voucher	7
3.7	LikeButton	7
3.8	HeurekaFeed	8
3.9	Menu	8
3.10	MSChart	8
3.11	FavouriteItems	8
3.12	HeurekaGetter	9
4	Postup řešení zadaných úkolů	10
4.1	Paragrafy	10
4.2	EmailChecker	10
4.3	XmlViewer	11
4.4	Xslt	11
4.5	Xslt z více souborů	12
4.6	Voucher	12
4.7	LikeButton	13
4.8	HeurekaFeed	13
4.9	Menu	13
4.10	MSChart	14
4.11	FavouriteItems	14
4.12	HeurekaGetter	14
5	Studiem získané znalosti uplatněné v průběhu odborné praxe	16
6	Scházející znalosti či dovednosti v průběhu odborné praxe	17
7	Závěr	18
8	Reference	19

Seznam obrázků

1	Ukázka tlačítka Like	7
2	Ukázka vytvořené aplikace	11
3	Komponenta pro generování slevových kupónů	12

Seznam výpisů zdrojového kódu

1	Ukázka struktury vstupního XML souboru	6
2	Příklad regulárního výrazu: § 12 odst. 1 zákona č. 231/2004 Sb.	10
3	Ukázka javascriptové funkce a tagu pro zobrazení tlačítka	13
4	Ukázka nastavení App.config pro zápis do log souboru	15

1 Úvod

V následujících kapitolách uvedu odborné zaměření firmy, ve které jsem bakalářskou praxi vykonával, mou pracovní pozici, popíši jednotlivé projekty, na kterých jsem v průběhu praxe pracoval a zvolený postup řešení. Pokusím se uvést znalosti, které jsem získal v průběhu bakalářského studia na Vysoké škole báňské - technické univerzitě a jejich využití v praxi ve firmě E LINKX a.s.. Dále popíši znalosti, které mi v průběhu praxe chyběly, zhodnotím znalosti a zkušenosti nabyté praxí a její přínos pro můj profesní růst.

2 Popis odborného zaměření firmy a pracovního zařazení studenta

2.1 Odborné zaměření firmy

Firma E LINKX a.s., která je členem skupiny eD' system Group, se orientuje na poskytování softwarových řešení svým zákazníkům. K poskytování služeb firma využívá produkty společnosti Microsoft a také produkty třetích stran[2]. V roce 2005 získala firma E LINKX a.s. ocenění Microsoft Gold Certified Partner. Zaměření firmy je už od jejího založení směřováno na vývoj a implementaci software v oblasti informačních systémů, ERP, B2B a B2C[1].

2.2 Pracovní zařazení studenta

Bakalářskou praxi jsem vykonával jako programátor webových i desktopových aplikací v jazyce C#. Na většině projektů jsem pracoval v týmu s Romanem Krpelem, který také absolvoval bakalářskou praxi. Naším konzultantem bakalářské práce byl pan Ing. Roman Hrdý.

3 Seznam úkolů v průběhu praxe a jejich přibližná časová náročnost

3.1 Paragrafy

Projekt Paragrafy vznikl z důvodu potřeby analyzovat text uložený v databázi, ve kterém se vyskytují právní předpisy. V každém textu bylo potřeba zjistit, zda obsahuje nějaký právní předpis. Vzory které se mohou vyskytnout byly předem definované a každý vzor mohl mít jinou prioritu vzhledem k ostatním vzorům. Každý nalezený vzor v textu bylo následně potřeba rozdělit na jednotlivé části, jako jsou například paragraf, odstavec, číslo, rok a sbírka. Takto nalezené informace se následně ukládaly do databáze.

Časová náročnost: 38 hodin.

3.2 EmailChecker

EmailChecker je DLL knihovna pro ověřování emailových adres. Požadavek byl, co nej-
přesněji zkontrolovat, jestli zadaná emailová adresa opravdu existuje.

Časová náročnost: 20 hodin.

3.3 XmlViewer

Úkolem bylo vytvořit znovupoužitelnou komponentu pro Windows Form aplikaci, jejímž cílem bude editovat XML soubor.

```
<CATALOG>
  <CD>
    <TITLE>Empire Burlesque</TITLE>
    <ARTIST>Bob Dylan</ARTIST>
    <COUNTRY>Australia</COUNTRY>
    <COMPANY>Columbia</COMPANY>
    <PRICE>10.90</PRICE>
    <YEAR type="System.DateTime">1985</YEAR>
  </CD>
  <CD>
    <TITLE>Hide your heart</TITLE>
    <ARTIST>Bonnie Tyler</ARTIST>
    <COUNTRY>GB</COUNTRY>
    <COMPANY>CBS Records</COMPANY>
    <PRICE>9.90</PRICE>
    <YEAR type="System.DateTime">1988</YEAR>
  </CD>
</CATALOG>
```

Výpis 1: Ukázka struktury vstupního XML souboru

Ten se měl předat komponentě společně s názvem kořenového uzlu a názvem atributu, který se má zobrazovat v prvku typu ListBox pro výběr uzlu, jenž chceme editovat.

Časová náročnost: 13 hodin.

3.4 Xslt

Cílem projektu bylo převést data exportovaná z jednoho internetového obchodu do struktury, která je vyhovující pro import dat v jiném internetovém obchodu. Exportovaná data i data pro import byla uložena v XML formátu. Transformace se měla spouštět jako konzolová aplikace. Důraz byl kladen na rychlost převodu z důvodu velkého množství položek.

Časová náročnost: 16 hodin.

3.5 Xslt z více souborů

Projekt je obdobou předchozího projektu Xslt, ale v tomto případě bylo z internetového obchodu exportováno více souborů se vzájemnými referencemi a mým úkolem bylo podle požadavků sloučit v nich obsažené informace do jednoho výstupního XML souboru.

Časová náročnost: 19 hodin.

3.6 Voucher

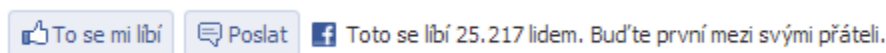
Smyslem úkolu bylo vytvořit do již hotové webové aplikace novou funkčnost, která umožní uživateli systému vygenerovat slevové kupóny. Nová webová stránka měla obsahovat formulář, ve kterém si přihlášený zástupce firmy zvolí druh slevy z možností poskytovaných pro jeho firmu a také počet vygenerovaných slevových kupónů. Uživatel s rozšířenými právy si bude moci navíc zvolit i firmu, pro kterou jsou slevové kupóny generovány. Takto vytvořené kupóny se následně uloží do databáze. Další část rozšíření, kterou vytvořil kolega, spočívala v zobrazování již vygenerovaných kupónů.

Časová náročnost: 62 hodin.

3.7 LikeButton

Úkolem bylo vytvořit WUC komponentu, která se umístí do webové stránky, rozpozná uživatele, který je přihlášen k Facebook účtu a umožní přímo ze stránky, kde je komponenta umístěna, přidat „Like“ aktuálně přihlášeného uživatele, nebo komentář k zobrazené stránce.

Časová náročnost: 7 hodin.



Obrázek 1: Ukázka tlačítka Like

3.8 HeurekaFeed

V projektu HeurekaFeed bylo společným úkolem vytvořit knihovnu, která z listu objektů reprezentujících produkt internetového obchodu vytvoří XML soubor v takovém tvaru, který je požadován pro import do informačního systému stránek Heureka¹ a Zboží². Já jsem pracoval na části pro import do informačního systému stránek Heureka, proto budu v této práci popisovat pouze tuto část.

Časová náročnost: 8 hodin.

3.9 Menu

Dynamické menu v informačním systému bylo již zastaralé. Bylo tvořeno pomocí zanořování div značek a stylem jednotlivých prvků, proto jej bylo nutno předělat, avšak vzhled a obsah musel zůstat stejný. Obsah menu byl načítán dynamicky z databáze. Nové menu mělo být vykreslováno pomocí značek ul a li.

Časová náročnost: 17 hodin.

3.10 MSChart

MS Chart Control³ je produkt společnosti Microsoft, který umožňuje vykreslovat grafy ve webových i desktopových aplikacích .NET Frameworku. Úkolem bylo si jej vyzkoušet a vykreslit 4 různé typy grafů.

Časová náročnost: 9 hodin.

3.11 FavouriteItems

Mým úkolem bylo změnit způsob přidávání oblíbených položek ve stávajícím informačním systému, které byly chybně navázány na položky, jenž se nadále mohly měnit. Proto bylo nezbytné tento systém předělat tak, aby změna původní položky neovlivňovala z ní vytvořenou oblíbenou položku.

Časová náročnost: 26 hodin.

¹<http://www.heureka.cz/>

²<http://www.zbozi.cz/>

³<http://msdn.microsoft.com/en-us/library/dd456632.aspx>

3.12 HeurekaGetter

Nákupní rádce Heureka obsahuje velké množství aktuálních cen různých produktů, které by se daly použít ke stanovení optimální ceny výrobku na trhu. Proto jsme dostali za úkol vytvořit DLL knihovnu, která podle vstupních parametrů vyhledá informace o daném produktu a uloží je do databáze. Další částí úkolu bylo vymyslet způsob, jak zamezit zablokování přístupu ze strany Heureka. Nastavení DLL knihovny se mělo provádět v souboru App.config a k ošetření výjimek a zápisu do log souboru jsme měli použít knihovnu Apache log4net⁴. K projektu jsme také měli vytvořit dokumentaci.

Časová náročnost: 98 hodin

⁴<http://logging.apache.org/log4net/>

4 Postup řešení zadaných úkolů

Postup řešení úkolů se v mnoha částech shodoval. V začátcích každého projektu bylo nezbytné se s projektem seznámit a vyhledat potřebné informace pro řešení projektu jako celku. Následoval pokus co nepřesněji odhadnout čas strávený na jednotlivých částech projektu (v hodinách). Poté započala první část implementace, po jejímž dokončení se upřesnil odhadovaný čas potřebný pro dokončení zbývajících částí projektu. Odhad času stráveného na následujících částech projektu se prováděl vždy po dokončení každé fáze (Iterační metodika vývoje). Konzultace s vedoucím se prováděla několikrát v průběhu projektu v závislosti na jeho složitosti.

4.1 Paragrafy

Vytvořil jsem konzolovou aplikaci, která načte řetězec z databáze a uloží je do slovníku. Poté se v cyklu prochází celý slovník a pro každý řetězec se zavolá metoda Vyhledej() z DLL knihovny. DLL knihovna obsahuje proměnné, ve kterých jsou uloženy řetězce s regulárními výrazy pro nalezení daného typu právního předpisu.⁵

```
private static string pattern = @"§()\s{1,2}(?<paragraf>\w+)\s{1,2}(odst\.\s{1,2}(?<odstavce>\w+)\s{0,2})(((?<podtyp>(zakon)|(vyhlask)|(narizeni)))\w*\s{1,2})č*(\.\s{1,2})?(?<cislo>\d+)/(?<rok>\d+)\s{0,2}(?<typ>((Sb\.\s{1,2})m\.\s{1,2})s\.\s{1,2})|(Sb\.\s{1,2}))?"
```

Výpis 2: Příklad regulárního výrazu: § 12 odst. 1 zákona č. 231/2004 Sb.

Pro každý regulární výraz obsahuje knihovna také metodu, jenž se stará o jeho zpracování a uložení do objektu. Metoda Vyhledej() obsahuje volání všech metod, které se starají o zpracování regulárního výrazu. Důležité je pořadí volání těchto metod. Obecnější regulární výrazy mohou obsahovat podmnožinu méně obecného regulárního výrazu, proto se musejí vyhledávat nejprve ty nejméně obecné výrazy. Pokud daná metoda najde shodu pro hledaný výraz, tak tuto část řetězce uloží do objektu, vymaže ze slovníku a pokračuje v hledání dalších typů vzorů. To zajistí, aby nebyla nalezena shoda pouze s částí řetězce pro obecnější regulární výraz. Po dokončení metody Vyhledej() se konzolové aplikaci vrátí všechny objekty reprezentující nalezené řetězce a voláním procedury se postupně uloží do databáze.

V prvních verzích aplikace jsem měl pouze několik velmi obecných typů právních předpisů a po zprovoznění aplikace a otestování funkčnosti jsem pokračoval v přidávání dalších a méně obecných typů předpisů. Úspěšnost nalezení správných řetězců se testovala již vytvořenou testovací aplikací.

4.2 EmailChecker

V DLL knihovně jsem kontroloval emailové adresy podle doporučení RFC 2822⁶ a RFC 3522⁷ za pomoci regulárního výrazu, dále pak existenci domény s použitím metody

⁵Příklad výpis 2 je uveden bez diakritiky.

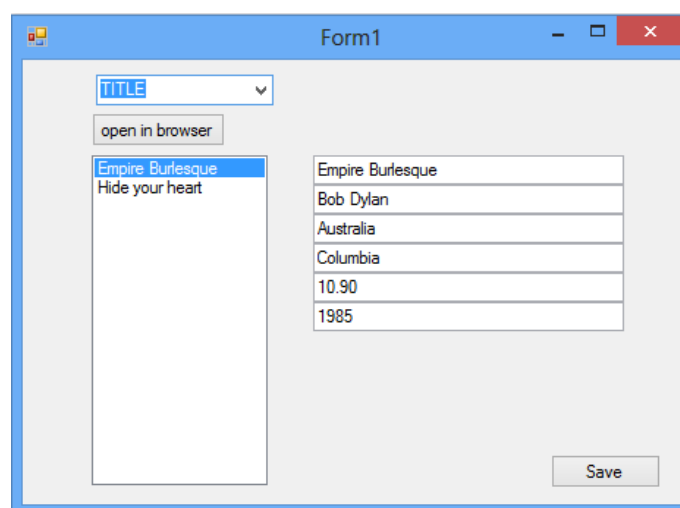
⁶Specifikace standardu na <http://tools.ietf.org/html/rfc2822>

⁷Specifikace standardu na <http://tools.ietf.org/html/rfc3522>

System.Net.Dns.GetHostEntry() a platnost celé emailové adresy jsem ověřoval za pomoci knihovny „JH Software’s DNS Client Library for .NET“⁸, která umožňuje získat z DNS serveru MX záznamy. S MX záznamem pak stačilo pouze vytvořit TCP spojení se serverem na port 25 a pokusit se odeslat zprávu „RCPT TO:“ se zadanou emailovou adresou. Server vrátí zprávu „250 2.1.5 Ok ...“ pokud obsahuje danou emailovou adresu. Některé servery ale tuto zprávu odesílají automaticky z důvodu bezpečnosti.

4.3 XmlViewer

Komponenta, kterou jsem vytvořil, přijímá v konstruktoru soubor typu XmlDocument, název kořenového uzlu a název atributu, který se má zobrazit. Pro testování jsem vytvořil Windows Forms aplikaci, která obsahovala pouze DropDownList s názvy elementů pro editaci. Po výběru se vygeneruje komponenta z vytvořené Windows Forms Control Library a zobrazí ListBox s hodnotami elementů, jenž byl předtím vybrán. Klikem na položku tohoto ListBoxu se zobrazí k ní příslušné hodnoty elementů v prvku typu Panel, který obsahuje dynamicky generované ListBoxy pro editaci a následně se stiskem tlačítka změny uloží.



Obrázek 2: Ukázka vytvořené aplikace

4.4 Xslt

V hlavní metodě programu bylo potřeba pouze vytvořit objekt třídy XslTransform a předat mu cestu ke vstupnímu, výstupnímu a transformačnímu souboru.

Všechnu ostatní logiku převodu jsem zapisoval do transformačního souboru s příponou xslt. Pro získání dat ze vstupního XML souboru jsem použil XPath dotazy, které podle logického stromu dokumentu naleznou požadovaný uzel[3]. Složitější práci s řetězci jsem

⁸Odkaz na stránky knihovny <http://www.simplifiedns.com/dns-client-lib.aspx>

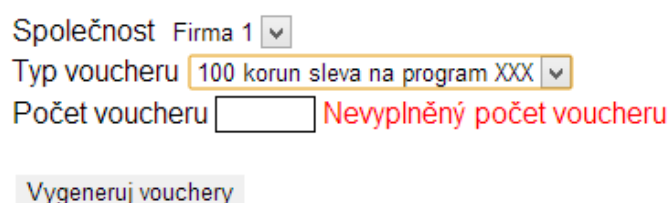
prováděl s pomocí vloženého skriptu v jazyce C#. K získání informací z řetězců jsem využil třídu Regex, ovšem z důvodu rychlosti bylo v některých případech výhodnější použít metody z třídy String.

4.5 Xslt z více souborů

Řešení této transformace se od projektu Xslt liší ve třídě, která se používá pro zpracování souborů. Pro tento projekt je to XslCompiledTransform. Transformace se vyvolá totožným způsobem, ale v tomto případě je vstupním souborem libovolný ze vstupních XML dokumentů. Proto jsem si zvolil ten s nejvíce daty a v samotném transformačním souboru se ostatní dokumenty načítají do proměnných. Z těch se následně získávají hodnoty dotazem podle jednoznačných klíčů, kterými jsou dokumenty vzájemně propojeny.

4.6 Voucher

Z důvodu zachování způsobu implementace informačního systému jsem vytvořil WUC komponentu, která je vložena do části pro samotný obsah stránky. Výhodou takového způsobu implementace je znovupoužitelnost kódu v celé aplikaci. Komponenta obsahuje DropDownList pro výběr společnosti, který má zapnutý AutoPostBack, ale pokud je do systému přihlášen zástupce konkrétní společnosti, tak se DropDownList do stránky vůbec nevygeneruje. Pro zjištění, jestli je v systému přihlášen zástupce společnosti slouží proměnná typu Boolean. Id uživatele se zdědí z rodičovského objektu. Pro zjištění správnosti zadaných hodnot jsem použil validační prvky, které ASP.NET poskytuje. Jsou jimi RequiredFieldValidator znázorněný na obrázku 3 a RegularExpressionValidator. Při odesílání stránky klientovi jsou tyto validační prvky přeměněny na JavaScript, který kontroluje správnost údajů přímo na straně klienta a odpadá tak zbytečná komunikace se serverem. Kontrolu zadaných vstupů je ovšem potřeba provádět i na straně serveru, protože uživatel může JavaScript vypnout a zamezit tak validování vstupů na své straně[4]. Komponenta dále obsahuje tlačítko třídy Button, jímž se vygenerují všechny požadované slevové kupóny, které se pomocí mapovacích objektů uloží do databáze.



Společnost Firma 1 ▼

Typ voucheru 100 korun sleva na program XXX ▼

Počet voucheru Newyplněný počet voucheru

Vygeneruj vouchery

Obrázek 3: Komponenta pro generování slevových kupónů

Mapovací objekty jsem společně s kolegou vytvořil podle již hotových objektů. Ke zjednodušení a urychlení implementace se ve firmě používá vlastní knihovna funkcí pro komunikaci s databází. Vytvoření a úpravu databázových tabulek zajistil kolega napsáním skriptu.

4.7 LikeButton

Facebook tlačítko jsem do stránky vykreslil za pomoci javascriptu a HTML5. Do komponenty jsem vložil javascriptovou funkci a tag pro zobrazení tlačítka podle návodu⁹.

```
<script type="text/javascript" language="javascript">
  (function (d, s, id) {
    var js, fjs = d.getElementsByTagName(s)[0];
    if (d.getElementById(id)) return;
    js = d.createElement(s); js.id = id;
    js.src = " //connect.facebook.net/en_US/all.js#xfbml=1";
    fjs.parentNode.insertBefore(js, fjs);
  } (document, 'script', 'facebook-jssdk'));
</script>
<div class="fb-like" data-send="true" data-width="450" data-show-faces="true"></div>
```

Výpis 3: Ukázka javascriptové funkce a tagu pro zobrazení tlačítka

Pro zobrazení komponenty ve stránce je potřeba JavaScript SDK¹⁰. Informace o stránce musí být obsaženy na začátku stránky v meta tagu. To byl problém, z důvodu jejich složitěho vypisování. Proto se ve funkci, která se spouští při vytvoření komponenty automaticky nastavují meta tagy. Obsah meta tagů se zadává do proměnných WUC komponenty při jejím použití.

4.8 HeurekaFeed

Vytvořil jsem DLL knihovnu obsahující dvě třídy. První reprezentuje produkt z Heureka a druhá představuje třídu pro vytvoření XML souboru. Parametrem předaným do konstruktoru je list produktů. Ten se následně prochází a do XML struktury se postupně přidává uzel s informacemi o produktu strukturovanými dle požadavků¹¹.

4.9 Menu

Menu jsem se rozhodl vytvořit s použitím Menu komponenty. Tato komponenta dokáže generovat a značky. Podle původního menu jsem si získal zanořený list položek z databáze a vyplnil jím Menu komponentu. Pak už jen stačilo nastavit kaskádové styly, aby vzhled menu odpovídal tomu původnímu. Menu jsem otestoval v poslední verzi prohlížečů: Google Chrome, Mozilla Firefox, Internet Explorer a Safari. Ve všech se komponenta vykreslila stejně kromě prohlížeče Safari, protože jinak počítá šířku prvku.

⁹Návod pro přidání facebook tlačítka na <https://developers.facebook.com/docs/reference/plugins/like/>

¹⁰<http://developers.facebook.com/docs/reference/javascript/>

¹¹Specifikace XML souboru <http://sluzby.heureka.cz/napoveda/xml-feed/>

4.10 MSChart

Stáhl jsem si a nainstaloval Microsoft Chart Controls for Microsoft .NET Framework 3.5¹². V designeru jsem vytvořil tři grafy. První měl spojnicí typu Spline a proložil body načtené z databáze. Druhý graf byl také typu Spline, ale navíc byl prostorový. A třetí graf byl prostorový, ale obsahoval dva typy dat, Spline a Column (proložená čára a sloupcový). Poslední graf typu Cylinder jsem si vyzkoušel naprogramovat ručně v C# kódu. Každému typu grafu a vyobrazení dat jsem změnil některé parametry a tak upravil jeho vzhled.

4.11 FavouriteItems

Nechtěl jsem dělat moc velké úpravy v databázové struktuře a business objektech, proto jsem se rozhodl pouze přidat sloupec, který bude rozhodovat, jestli je položka určena pro normální nebo oblíbenou skupinu. To však bylo špatně. Z důvodu optimalizace rychlosti systému je mnohem lepší vytvořit novou tabulku, aby se urychlilo vyhledávání a také aby databázová struktura byla přehlednější. Se změnou struktury databázových tabulek bylo také potřeba upravit business objekty pro načítání dat z databáze a pozměnit databázový pohled. Pohled byl použit pro spojení detailních informací o dané položce s tabulkou ukazující na jejich použití pro snazší načítání dat do objektů. Následně jsem upravil WUC komponenty, které zobrazovaly upravené databázové struktury.

4.12 HeurekaGetter

Vytvořil jsem DLL knihovnu, třídu pro načítání dat z heureka a třídu reprezentující produkt. Pro získávání informací ze staženého HTML dokumentu jsem použil knihovnu HtmlAgilityPack¹³, která se používá podobně jako třída XmlDocument a kolega vytvořil metody pro získávání jednotlivých informací o produktech. Po zjištění, že Heureka obsahuje dva způsoby vyhledávání produktu (vyhledávání ve všech položkách a vyhledávání v konkrétních kategoriích) jsem vytvořil abstraktní třídu, která poskytuje společné virtuální metody a proměnné pro vyhledávání. Dál se ovšem pracovalo pouze na vyhledávání ve všech položkách. K testování vytvořil kolega uživatelské rozhraní. Zaznamenávání do log souboru jsem zajistil nainstalováním knihovny log4net za pomoci rozšíření NuGet package manager¹⁴, nastavením souboru App.config pro zápis do konzole, do souboru viz výpis 4 i do databáze a voláním funkcí log4net z kódu programu, kde jsem odchytil výjimky.

¹²Odkaz pro stažení MS Chart Controls <http://www.microsoft.com/en-us/download/details.aspx?id=14422>

¹³Dostupné z <http://htmlagilitypack.codeplex.com/>

¹⁴Více informací o NuGet na <http://nuget.org/>

```
<log4net>
  <root>
    <appender-ref ref="FileAppender"/>
  </root>
  <appender name="FileAppender" type="log4net.Appender.FileAppender">
    <file value="log-file.txt" />
    <appendToFile value="true" />
    <lockingModel type="log4net.Appender.FileAppender+MinimalLock" />
    <layout type="log4net.Layout.PatternLayout">
      <conversionPattern value="%date_[%thread]_-5level_%-logger_-%message%newline%
        exception" />
    </layout>
  </appender>
</log4net>
```

Výpis 4: Ukázka nastavení App.config pro zápis do log souboru

K vytvoření dokumentace jsem použil nástroj pro generování dokumentace Doxygen¹⁵, kterým jsem vygeneroval soubor s příponou RTF. Vygenerovaný soubor jsem následně doplnil informacemi pro použití knihovny a ukázkami nastavení v souboru App.config.

¹⁵Stránky projektu Doxygen na <http://www.stack.nl/~dimitri/doxygen/index.html>

5 Studium získané znalosti uplatněné v průběhu odborné praxe

V průběhu odborné praxe jsem uplatnil znalosti získané během studia od prvního až do třetího ročníku. Po celou dobu praxe, kdy jsem převážně programoval v jazyce C#, jsem těžil ze znalostí objektově orientovaného programování, které jsem se učil v předmětu Algoritmy. Nepostradatelné pro mne byly znalosti o tvorbě databázových struktur z Úvodu do databázových technologií a také znalost jazyka T-SQL z předmětu Databázové a informační systémy. Při práci s webovou aplikací v projektu Voucher jsem čerpal z informací získaných v předmětu Vývoj informačních systémů, kde mi byla výhodou znalost způsobu přístupu k datům uloženým v databázi a základy platformy ASP.NET. Velkou měrou jsem čerpal z informací získaných v předmětu Programovací jazyky, které se řadí mezi nejvýznamnější především díky osvojení syntaxe jazyka C# používaného v průběhu celé praxe, práci s XML, kolekcemi, ošetřování výjimek a mnoha dalších. Přínosem pro mě byly i znalosti z předmětu Vývoj internetových aplikací o HTML a CSS stylech a informace o mailových serverech získané v předmětu Počítačové sítě, uplatněné v projektu EmailChecker. Použil jsem také informace o UML diagramech osvojené v hodinách softwarového inženýrství.

Mezi dovednosti získané v průběhu studia rozhodně patří práce s nástroji Microsoft Visual Studio, Microsoft SQL Management a Microsoft Visio.

6 Scházející znalosti či dovednosti v průběhu odborné praxe

Vzhledem k zaměření firmy E LINKX a.s. na produkty společnosti Microsoft, které máme na naší fakultě k dispozici a k velmi dobré přípravě ze strany školy, jsem se za celou dobu odborné praxe nepotýkal s žádným zásadním nedostatkem informací. Novinkou pro mě byl responzivní design, který se používá k nastavení vzhledu webových stránek pro různé velikosti zobrazovacích zařízení. Z počátku pro mne byla nedostatečná hloubka praktických informací především v oblasti technologie ASP.NET, to se však v průběhu intenzivní práce na projektech zlepšilo.

7 Závěr

Odborná bakalářská praxe pro mne byla neocenitelným přínosem zejména díky nabytým zkušenostem, praktickým informacím a dovednostem, které jsem si v jejím průběhu osvojil. Získal jsem mnoho informací ohledně vývoje softwaru ve firemních podmínkách a způsobech implementace. Přínosem pro mě byla také práce v týmu na některých projektech. Díky odborné bakalářské práci jsem si výrazně prohloubil znalosti technologie .NET a mnoha jejích tříd. Bakalářská praxe představuje důležitou část mého profesního života z pohledu kariérního růstu, především kvůli důležitosti praktických zkušeností pro uplatnění na trhu práce v oblasti informačních technologií a vazbě na firemní sektor.

8 Reference

- [1] E LINKX a.s.: Jsme vždy dál. Nedržíme pouze krok s vývojem. E LINKX a.s. [online]. 1999 - 2013 [cit. 2013-04-14]. Dostupné z: <http://www.elinkx.cz/pagelist.aspx?eid=835>
- [2] E LINKX a.s.: O společnosti. E LINKX a.s. [online]. 1999 - 2013 [cit. 2013-04-14]. Dostupné z: <http://www.elinkx.cz/o-spolecnosti>
- [3] SKONNARD, Aaron a Martin GUDGIN. XML - pohotová referenční příručka: referenční příručka programátora ke XML, XPath, XSLT, XML Schema, SOAP a dalším. 1 vyd. Překlad Lucie Rút Bittnerová. Praha: Grada, 2006, 342 s. ISBN 80-247-0972-4.
- [4] PÍSEK, Slavoj. ASP.NET: začínáme programovat : podrobný průvodce začínajícího uživatele. 1. vyd. Praha: Grada, 2003, 228 s. ISBN 80-247-0526-5.